IMPROVE NOVEL CLASS GENERALIZATION BY ADAPTIVE FEATURE DISTRIBUTION FOR FEW-SHOT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

In this work, we focus on improving the novel class generalization of few-shot learning. By addressing the difference between feature distributions of base and novel classes, we propose the adaptive feature distribution method which is to finetune one scale vector using the support set of novel classes. The scale vector is applied on the normalized feature distribution and by using one scale vector to reshape the feature space manifold, we obtain consistent performance improvement for both in-domain and cross-domain evaluations. By simply finetuning one scale vector using 5 images, we observe a 2.23% performance boost on 5-way 1-shot cross-domain evaluation with CUB over statistics results of 2000 episodes. This approach is simple yet effective. By just finetuning a single scale vector we provide a solution of reducing number of parameters while still obtain generalization ability for few-shot learning. We achieve the state-of-the-art performance on mini-Imagenet, tiered-Imagenet as well as cross-domain evaluation on CUB.

1 INTRODUCTION

With the plethora of available large-scale data, deep learning has achieved significant advancements. However multiple factors such as high labelling costs, scarce availability of classes of interest or the expensive need for experts for label generation set limits of applying large-scale data. To address this challenge, the problem of few-shot learning was formulated which has received considerable attention in recent years Vinyals et al. (2016); Snell et al. (2017); Finn et al. (2017); Ravi & Larochelle (2016); Hariharan & Girshick (2017).

For a supervised learning problem with data set $(x_1, y_1), ...(x_n, y_n)(x_i \in \mathcal{X}$ feature space, $y_i \in \mathcal{Y}$ label space), by using the hypothesis class($h(.; \mathbf{w})$), we want to minimize $l(h(x; \mathbf{w}), y)$ on new samples. *With the assumption that training samples and test samples are i.i.d from the same unknown distribution D over $\mathcal{X} \times \mathcal{Y}$*, the problem is optimized over the Empirical Risk Minimization(ERM). For multi-class classification with deep neural network, the hypothesis class related to the scenario can be divided into two functionalities: the feature extractor $F_\theta(x_i)$ parameterized by θ, the classifier $C(\cdot \mid \mathbf{w})$ for a given class weight vector $\mathbf{w}$. Basically to achieve a good classification performance over the large-scale dataset, $h(.; \mathbf{w})$ is expected to be highly invariant and this property empowers the feature extractor $F_\theta(x_i)$ with good feature invariance ability if we consider variations that are generally in the objects such as shapes, lights and etc.

Few-shot learning proposes a great challenge as the estimation of the distribution is hard to achieve with a few samples. Meta-learning methods on few-shot learning lead a direction of adapting to a hypothesis class with few samples, which directly back-propagates the loss between testing set with the $h(.; \mathbf{w})$ proposed with the training set. Recent work meta-Baseline Chen et al. (2020)proposed to conducts meta-training with a pre-trained feature extractor on base classes which leads to a large-margin performance improvement of meta-training. Moreover, they observe that during meta-training stage, models are better generalized on the base classes while evaluation performance on novel classes largely dropped.

The novel class generalization which is defined as evaluation performance on novel classes following Chen et al. (2020) is essential for improving few-shot learning into practice. Training of algorithms on few-shot learning are conducted with base classes which are relatively large-scale in the sense

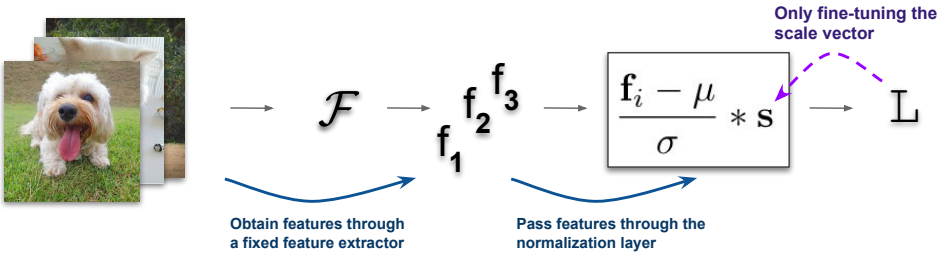


Figure 1: Illustration on AFD: with samples from support sets of novel classes, features are obtained by using a pre-trained feature extractor $\mathcal{F}$ and then these features are passed through the feature normalization layer which is parameterized by a scale vector; with using a non-parametric evaluation metrics gradients flow into optimizing the scale vector.

of plenty number of classes with hundreds of images. Methods in metric-based learning Chen et al. (2019); Gidaris & Komodakis (2019); Wang et al. (2018); Gidaris & Komodakis (2018) and meta-learning Chen et al. (2020) prove that training in this way benefits the capture of large variations which is crucial for discriminative features. However, as the feature extractor on base classes is trained under maximum likelihood, features are also trained to be invariant for discriminating these base classes, as shown in Fig.2. Then the evaluation on novel classes would suffer from the feature distribution difference between base and novel classes, and cross-domain between base and novel classes could enlarge this feature distribution difference. Objects(or images) in different domains carry different aspects of information which leads to different discriminative features or features in common among categories.

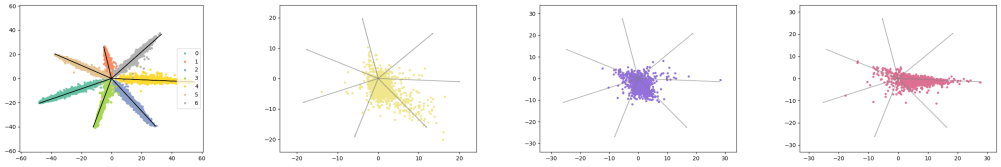
Attempts of improving novel class generalization include finetuning method proposed in Chen et al. (2019). In Chen et al. (2019), they proposed to finetune the novel class weights using the support set of novel classes with competitive results. However if feature distribution of novel classes suffers from scattering, even with a plenty of data finetuning the novel class weights without any optimization on the feature side is not promising for finding a good decision boundary, not to mention with only a few samples.

In our work, we propose the adaptive feature distribution to improve the novel class generalization. Following the idea of finetuning using a handful of samples, we apply a non-parametric distance first to construct the hypothesis class and then by only finetuning a scale vector which applied on the normalized feature distribution, we achieve the effects of adaptive feature distribution on novel classes.

Our Contributions: 1) We address the importance of further understanding the feature distribution for novel classes. Using DB-index which measures the quality of feature distributions for novel classes to select feature extractors, we observe a consistent performance boost on all three evaluation datasets. We believe introducing analysis on feature distributions and clustering quality of novel classes is informative to the community. **2)** We propose to improve novel class generalization through adapting the feature distribution of novel classes. And by only finetuning one scale vector using support sets of novel classes, we showcase the supreme generalization of this method especially on cross-domain evaluations. We achieve the state-of-the-art performance on mini-Imagenet, tiered-Imagenet as well as cross-domain evaluation on CUB. **3)** This approach is simple yet effective. By just finetuning a single scale vector we provide a solution of reducing number of parameters while still obtain generalization ability for few-shot learning.

2 Prior Art

There have been many approaches to few-shot learning explored recently, namely are fast-adaptation methods Finn et al. (2017); Rusu et al. (2018); Sun et al. (2019); Chen et al. (2020), model optimization based methods Ravi & Larochelle (2016), metric learning based methods Vinyals et al. (2016); Snell et al. (2017); Ren et al. (2018); Sung et al. (2018); Guo & Cheung (2020); Li et al. (2020) and methods which use ridge regression and support vector machine Bertinetto et al. (2018); Lee et al.



(a) Learned Representation for Base Classes (b) Novel Class 7 (c) Novel Class 8 (d) Novel Class 9

Figure 2: MNIST Illustration of Feature Distribution Difference between Base and Novel Classes. The feature extractor(Lenet) is trained with 0-6 base classes. We plot the feature distribution for base classes and novel classes. As shown in 2-D space, novel classes features are more scattered compared with compact feature distribution of base classes. Meanwhile, novel class features tend to project on the direction of base class weights(shown as the gray line).

(2019). There have also been studies focusing on discovering projective feature embeddings Simon et al. (2018; 2020). Recently, a few studies utilized a variety of techniques in machine learning towards few shot classification. Techniques like self-supervised training, self-training through semi-supervised learning and model ensembles showed a boost result when applied on few-shot learning problem Gidaris et al. (2019); Dvornik et al. (2019); Li et al. (2019b). Modules were also invented to enhance feature discrimination Li et al. (2019a); Hou et al. (2019). Recently approaches have also explored combination with Graph Neural Networks Garcia & Bruna (2017); Kim et al. (2019).

3 Adaptive Feature Distribution: Fine-tuning the Scale of Feature Distribution on Novel Classes

In this section, we introduce how we realize the adaptive feature distribution with a learnable scale vector and the effects on novel class feature space by only finetuning the scale vector with a few samples.

The Few-Shot Problem Formulation. Evaluation datasets in few-shot learning are separated into base, validation and test classes. *Base classes* which is used in training involves a relatively large number of labelled training samples. And *validation classes* or *test classes* are treated as *novel classes*, which correspondingly used for validation and testing purpose. For few-shot learning scenarios, one episode is defined as K-way N-shot learning where K is the number of classes, N is the number of training images(support set) and K classes are firstly sampled from the novel classes; N samples in the support set as well as the query set(samples used for evaluating the episode performance) are sampled within each K classes. For one K-way N-shot episode, we use S_k and Q_k to denote the support and query set accordingly for $k \in K$ novel classes.

We use a pre-trained feature extractor F_θ to subtract features. We use $\mathbf{f}_i = F_\theta(\mathbf{x}_i)$ to represent the feature for x_i. We first add a feature normalization layer with a scale vector $\mathbf{s}$:

$$\bar{\mathbf{f}}_i = \frac{\mathbf{f}_i - \mu}{\sigma} * \mathbf{s} \tag{1}$$

Where: $\mu = \frac{1}{N}\sum_i \mathbf{f}_i$ and $\sigma = \frac{1}{N}\sum_i (\mathbf{f}_i - \mu)^2$.

In this layer, features from all training samples are first normalized in a way that values of every element on the feature vectors are regularized by following the normal distribution. A scale vector $\mathbf{s}$ is then multiplied with the normalized feature. $\mathbf{s}$ serves as the "adaptive" part that by tuning the value of $\mathbf{s}$, we are scaling the normalized feature distribution. $\mathbf{s}$ is flexible in the sense that every element on $\mathbf{s}$ scales up or down on every element of features and this in general leads to the reshape of feature space manifold. Then by fine-tuning $\mathbf{s}$ with classification loss on novel classes, we expect to the reshape of feature space manifold could fast adapt the features for novel classes especially on cross-domain cases.

In the fine-tuning stage, we first construct our evaluation metrics in an non-parametric way. We use average feature of the support set S_k as the class weight w_k with a softmax loss:

$$L_f = -\frac{1}{N}\sum_{i=1}^{N}\log P_{y_i} = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{\exp z_{y_i}}{\sum_{k=1}^{K}\exp z_k} \tag{2}$$

Where

$$z_j = \mathbf{w}_j^T \cdot \bar{\mathbf{f}}_i = \bar{\mathbf{f}}_i^T \cdot \frac{1}{N}\sum_{\mathbf{x}\in S_j}\bar{\mathbf{f}} \tag{3}$$

By using this non-parametric metrics, we decrease the number of parameters to be trained in the fine-tuning stage while still follows the maximum likelihood estimation to predict the probability $p(y|x)$. And this allows flexibility of fine-tuning the feature space with adaptive feature distribution. We analyze the gradient flow in the fine-tuning stage in the following.

The derivative ofz_j with $\bar{\mathbf{f}}_i$ is:

$$\frac{\partial z_j}{\partial \bar{\mathbf{f}}_i} = \mathbf{w}_j \tag{4}$$

For an input x_i, the derivative of z_j with L_f is:

$$\frac{\partial L_f}{\partial z_j} = \begin{cases} P_j - 1 & j = y_i \\ P_j & j \neq y_i \end{cases} \tag{5}$$

$$\frac{\partial L_f}{\partial \bar{\mathbf{f}}_i} = (P_{y_i} - 1)\mathbf{w}_{y_i} + \sum_{j\neq y_i}^{K} P_j \mathbf{w}_j \tag{6}$$

Meanwhile as s is element-wisely multiplied with $\mathbf{f}_i$, the gradient at location c for $\mathbf{s}$ is(we omit the notation of location c to simplify the notation):

$$\frac{\partial \bar{f}_i}{\partial s} = \frac{f_i - \mu}{\sigma} \tag{7}$$

Then we have the gradient for s at any location on $\mathbf{s}$ with sample x_i as:

$$\nabla s = \frac{f_i - \mu}{\sigma}[(P_{y_i} - 1)w_{y_i} + \sum_{j\neq y_i}^{K} P_j w_j] \tag{8}$$

For fine-tuning only using K-way N-shot samples, $P_{y_i} \simeq 1$(for 1-shot case, $P_{y_i} = 1$) the gradient during training can be approximated as:

$$\nabla s = \frac{f_i - \mu}{\sigma}\sum_{j\neq y_i}^{K} P_j w_j = \frac{f_i - \mu}{\sigma}\sum_{j\neq y_i}^{K}[P_j \sum_{x\in S_j} f_x] \tag{9}$$

To simplify the notation, we use the gradient for 1-shot case to conduct further discussion, which is:

$$\nabla s = \frac{f_i - \mu}{\sigma}\sum_{j\neq y_i}^{K} P_j \frac{f_j - \mu}{\sigma} \tag{10}$$

By conducting the gradient descent, we have $s = s - \nabla s$.

To give a direct impression of how this fine-tuning changes the feature space manifold, we illustrate the change on s brought by gradient descent intuitively. First of all, the normalization on f ensures that the value is "soft bounded", which will not cause the extreme values on the gradient. For some locations where elements are encoded "common" information, values of f_i and f_j are similar. And in the opposite way, elements in other locations are encoded "discriminative" information where values of f_i and f_j are largely different. In this case, ∇s could be relatively large or negative which leads to scaling up the feature distribution at those locations. Then the difference between features are further enlarged correspondingly. In this case, the manifold of the feature space will fast adapt to the shape where distinguished parts are enlarged.

4 OVERALL FRAMEWORK

In this section, we introduce the overall framework that we conduct the few-shot classification problem.

4.1 TRAINING CLASSIFICATION ON BASE CLASSES

The model $F_\theta(\mathbf{x})$ that is trained on the base classes K_{base}. To obtain a better feature invariance, we use the l2-normalized Softmax Chen et al. (2019); Ranjan et al. (2017); Wang et al. (2017); Qi et al. (2018) with cross entropy loss, which utilize softmax under the constraint of $\|\mathbf{w}_{y_i}\|_2^2 = 1$ and $\|F_\theta(\mathbf{x}_i)\|_2^2 = 1$:

$$L_{SM} = -\frac{1}{N_{base}} \sum_{i=1}^{N_{base}} \log \frac{\exp S\cos(\mathbf{w}_{y_i}^T, F_\theta(\mathbf{x}_i))}{\sum_{k=1}^{K_{base}} \exp S\cos(\mathbf{w}_k^T, F_\theta(\mathbf{x}_i))} \tag{11}$$

4.2 EVALUATION ON NOVEL CLASSES

Given an K-way N-shot episode of few-shot classification, for each class $k \in K$ we have a support set $S_k = (x_1, y_1), \cdot, (x_N, y_N)$ and a query set $Q_k = (x_1, y_1), \cdot, (x_M, y_M)$. With the pretrained feature extractor $F_\theta(\mathbf{x})$, we follow the same metric of cosine distance in equation 11 when evaluating on novel classes; and the novel class weight w_k is the average feature of the support set S_k Qi et al. (2018); Chen et al. (2020):

$$\mathbf{w}_k = \frac{1}{N} \sum_{\mathbf{x} \in S_k} F_\theta(\mathbf{x}) \tag{12}$$

The predicted probability that $\mathbf{x} \in Q$ belongs to class k is:

$$p(y = k|\mathbf{x}) = \frac{\exp\cos(\mathbf{w}_k^T, F_\theta(\mathbf{x}))}{\sum_{j=1}^{K} \exp\cos(\mathbf{w}_j^T, F_\theta(\mathbf{x}))} \tag{13}$$

4.3 FINE-TUNING SCALE VECTOR ON NORMALIZED FEATURE DISTRIBUTION.

For the fine-tuning part, we conduct experiments with data augmentation and without data augmentation separately. With data augmentation, when we construct our the non-parametric evaluation metrics in equation. 2 the average feature used for novel class weight are generated from samples without data augmentation while features as input to the evaluation metrics are from samples after data augmentation. By doing this, we ensures the minimum change of the novel class prototype(class weight) and the maximum of sample variations around the class prototype. The fine-tuning without data augmentation follows the methodology in Section 2.

4.4 MODEL SELECTION FOR NOVEL CLASSES.

After we train the classification on base classes, we come to the model selection of using which model as the feature extractor for novel classes. The feature extractor with the best classification accuracy or from the later epochs may not be a good choice. To obtain a high classification accuracy, features trained by supervised classification at the later stage of training may suffer the "overfitting" to the seen classes. In other words, features would be projected precisely to directions of class weight vectors in order to get a high classification accuracy. By using these models as the feature extractor for novel classes, features of the novel classes could be separately projected onto the directions of the base classes which enlarge the scattering of that feature distribution indeed. Using the few-shot performance on validation set could be one choice, however as we are approaching the adaptive feature distribution, we consider the model selection from the perspective of measuring the quality of feature distribution. We use DB-index Davies & Bouldin (1979) as the criterion for model selection, which evaluates the clustering quality by considering the separation of the clusters and the tightness inside the clusters. And interestingly, we found that models with lower DB-index are generally models around the epoch after the first time of decreasing the learning rate. In our experiments, models with lower DB-index on validation set are selected.

Models	Backbone	mini-ImageNet		tiered-ImageNet	
		1-shot	5-shot	1-shot	5-shot
Finn et al. (2017)	Conv-4-64	48.70 ± 1.84	63.10 ± 0.92	51.67 ± 1.81	70.30 ± 0.08
Sung et al. (2018)	Conv-4-64	50.44 ± 0.82	65.32 ± 0.70	-	-
Gidaris et al. (2019)	WRN-28-10	62.93 ± 0.45	79.87 ± 0.33	**70.53** ± 0.51	84.98 ± 0.36
Gidaris & Komodakis (2019)	WRN-28-10	61.07 ± 0.15	76.75 ± 0.10	68.18 ± 0.16	83.09 ± 0.12
Rusu et al. (2018)	WRN-28-10	61.76 ± 0.08	77.59 ± 0.12	66.33 ± 0.05	81.44 ± 0.09
Gidaris & Komodakis (2019)	WRN-28-10	60.06 ± 0.14	76.39 ± 0.11	68.18 ± 0.16	83.09 ± 0.12
Li et al. (2019a)	ResNet18	62.05 ± 0.55	78.63 ± 0.06	64.78 ± 0.11	81.05 ± 0.52
Dvornik et al. (2019)	ResNet18	59.48 ± 0.62	75.62 ± 0.48	70.44 ± 0.32	**85.43** ± 0.21
Oreshkin et al. (2018)	ResNet12	58.50 ± 0.30	76.70 ± 0.30	-	-
Ravichandran et al. (2019)	ResNet-12	60.71	77.26	66.87	82.64
Lee et al. (2019)	ResNet12	62.64 ± 0.61	78.63 ± 0.46	65.99 ± 0.72	81.56 ± 0.53
Sun et al. (2019)	ResNet12	61.2 ± 1.8	75.5 ± 0.8	-	-
Simon et al. (2020)	ResNet-12	**64.60** ± 0.72	79.51 ± 0.50	67.39 ± 0.82	82.85 ± 0.56
Guo & Cheung (2020)	ResNet-12	63.12 ± 0.08	78.40 ± 0.11	67.69 ± 0.11	82.82 ± 0.13
Li et al. (2020)	ResNet-12	-	-	67.10 ± 0.52	79.54 ± 0.60
Chen et al. (2020)	ResNet-12	63.17 ± 0.23	79.26 ± 0.17	68.62 ± 0.27	83.29 ± 0.18
Baseline	ResNet12	59.38 ± 0.44	76.83 ± 0.33	63.51 ± 0.48	80.46 ± 0.38
Baseline*	ResNet12	**63.73** ± 0.44	80.59 ± 0.31	68.68 ± 0.49	84.03 ± 0.35
AFD	ResNet12	63.70 ± 0.44	**80.81** ± 0.31	**68.72** ± 0.49	**84.23** ± 0.35

Table 1: Results on mini-ImageNet and tiered-ImageNet for 5-way evaluation. The results are the average accuracy with 95% confidence intervals based on the same 2000 test episodes among all our experiments. The 95% confidence intervals is reference to comparing with other methods.

5 Experimental Validation

We evaluate the our adaptive feature distribution method in both in-domain case and cross-domain case. In-domain case is defined as the base and novel classes are from the same datsets and cross-domain case refers to the situation that base and novel classes are from different datasets and generally the datasets have domain difference.

5.1 Evaluation Datasets and Implementation Details

5.1.1 Evaluation Datasets

Dataset 1: mini-ImageNet Vinyals et al. (2016) is a standard benchmark for few-shot image classification benchmark, which consists of 100 randomly chosen classes from ILSVRC-2012 Russakovsky et al. (2015). And these classes are randomly split into 64, 16 and 20 classes for meta-training, meta-validation and meta-test set respectively. Each class contains 600 images of size 84×84. We use the common split used in Lee et al. (2019).

Dataset 2: tiered-ImageNet Ren et al. (2018) is a larger subset of ILSVRC-2012 Russakovsky et al. (2015), composed of 608 classes which are split into meta-training, meta-validation and meta-testing set with 351, 97 and 160 classes respectively. All images are of the size 84×84.

Dataset 3: CUB-200-2011 Wah et al. (2011) contains 200 classes and 11,788 images in total. Following the evaluation protocol of Hilliard et al. (2018), the dataset is split into 100 base, 50 validation and 50 novel classes. We use the same splits as Chen et al. (2019) for testing. This dataset serves as the test set for the cross-domain evaluation.

5.1.2 Architecture and Training Details

Baseline Network Architecture. We utilize the ResNet-12 network architecture following Lee et al. (2019) to train the baseline and backbone classification model. However in contrast to Lee et al. (2019), we use a global average pooling after the last residual block following which the feature length becomes 640 and the feature layer is followed by a 1-d batchnorm layer without affine.

method	5-way 1-shot	5-way 5-shot
MatchingNetsVinyals et al. (2016)	-	53.07 ± 0.74
MAMLFinn et al. (2017)	-	51.34 ± 0.72
ProtoNetSnell et al. (2017)	-	62.02 ± 0.70
Linear Classifier(Chen et al. (2019))	-	65.57 ± 0.7
Cosine Classifier(Chen et al. (2019))	-	62.04 ± 0.76
Diverse 20 FullDvornik et al. (2019)	-	66.17 ± 0.55
Baseline	46.31 ± 0.43	64.15 ± 0.38
Baseline*	49.26 ± 0.43	69.56 ± 0.39
AFD	**50.99** ± 0.43	**70.64** ± 0.38

Table 2: Domain Difference Testing on CUB Dataset using the mini-ImageNet Trained Model. Results for MatchingNets, MAML and ProtoNet are fromChen et al. (2019).

Training hyperparameters. All networks were trained with SGD along with Nesterov momentum of 0.9 and weight decay of 5×10^{-4}. The initial learning rate is set as 0.1 which was decreased by a factor of 10 every 50 epochs for a total of 150 epochs. The batch size was kept at 256. Data argumentation was applied for baseline classification following Lee et al. (2019), which included horizontal flip, random crop, and color (brightness, contrast, and saturation) jitter.

Fine Tuning on the Novel Class Support Set. We finetune on the novel class training set using Adam with learning rate 5×10^{-3} by back-propagating the gradient from the whole batch, and early stop in this case is crucial that we finetune 3 epochs for 1-shot and 5 epochs for 5-shot case in cross-domain cases and 3 epochs for both 1-shot and 5-shot in in-domain cases. The scale vector is initialized as 1.

In our experiments, *Baseline* refers to the pre-trained feature extractor of the last epoch for training; *Baseline** refers to the pre-trained feature extractor selected using density based clustering index. And we use *Baseline** as the feature extractor for all our finetuning experiments.

5.2 Comparing Performance on In-Domain and Cross-Domain Cases

Performance of model selection are consistent. We observe that using the DB-index to select feature extractors gain consistent performance improvement among all three evaluation datasets. And this could serve as a good sign of studying the feature transfer-ability from the perspective of feature distribution.

AFD Shows Improvement on In-Domain Evaluations. Shown in Table.1, AFD improves the performance on 5-shot with 0.22% and 0.2% separately for miniImagenet and tieredImagenet. One thing to notice is that the performance of our Baseline* already surpass performance of most works. AFD still leads to performance improvement while using a well presumed feature extractor.

AFD shows superior generalization to cross-domain evaluations. Shown in Table.2, by simply finetuning using 5 images for 1-shot case and 25 images for 5-shot case, we observe 1.73% and 1.08% performance improvement from statistical results among 2000 episodes.

5.3 Ablation Studies on Fine-tuning

The results of ablation studies are shown in Table.3.

Effects of Applying Data Augmentation during Finetuning: The major obstacle of few-shot learning is the lack of samples which is essential for improving the novel class generalization. Although we only train 3 epochs, the effects of data augmentation are still obvious. Only for 1-shot case with miniImagenet-trained feature extractor the performance is worse than without using data augmentation. This could be caused with the reason that the feature extractor is trained with a relatively small data, features abstracted then are not stable to optimize which is serve when adding data augmentation with only 5 training samples. Otherwise, we observe performance improvement of 0.47% for 5-shot with miniImagenet-trained feature extractor and 0.18%, 0.65% for 1-shot and 5-shot with tieredImagenet-trained feature extractor. As we only use the basically simple data aug-

Models	Components			mini-ImageNet		tiered-ImageNet	
	dot-product	cosine	data-aug	1-shot	5-shot	1-shot	5-shot
Baseline				46.31	64.15	46.52	65.59
Baseline*				49.26	69.56	54.67	74.94
finetune-weight		✓	✓	48.60	68.64	54.25	74.87
	✓			**51.49**	70.17	55.00	74.56
		✓	✓	50.99	70.60	55.18	75.21
AFD	✓		✓	50.99	**70.64**	**55.18**	**75.21**

Table 3: Ablation Studies on CUB. All experiments are in 5-way evaluations. The results are the average accuracy based on 2000 test episodes. Episodes are the same over all experiments. The 95% confidence intervals are approximately the same(with 0.01 difference among experiments), and we put the values correspondingly: 0.43, 0.38, 0.48, 0.39.

mentation strategy, further work to explore the effective data augmentation for finetuning on novel classes are promising.

Effects of Different Feature Extractor: Firstly by using a feature extractor trained with larger dataset, the performance on cross-domain cases boost a lot which indicates the importance of a good feature embedding. And for AFD, the performance improvement on 1-shot are 1.73% and 1.08% for miniImagenet model and tieredImagenet model; on 5-shot are 0.51% and 0.27% for miniImagenet model and tieredImagenet model. This illustrates that AFD are able to fast adapt features especially when the quality of feature embedding is not good. However, with a better feature extractor allows better improvement of using data augmentation in AFD as discussed above.

Effects of Different Metrics: We compare the results of using different metrics(dot-product and cosine metrics with scale Wang et al. (2017)) in our non-parametric evaluation for fine-tuning. The performance is almost the same. As different metrics affect how well can we achive the predicted probability and in our case, as illstrated in Section 2, the predicted probability is around 1 already. Then different metrics serve similar efforts of adapting features for novel classes.

The Importance of Fine-tuning Features: We compare AFD with only finetuning the novel class weight method. For finetuning the novel class weight, we use average features from support set as the weight initialization and the hyper-parameters settings are the same as mentioned above. We observe performance drop by only finetuning the novel class weight, which are 0.66%, 0.92% for 1-shot and 5-shot with mini-ImageNet trained feature extractor, 0.42%,0.07% for 1-shot and 5-shot with tiered-ImageNet trained feature extractor. For cross-domain cases, features for novel classes are not well discriminative and constrained for the same class. As features are not optimized, only finetuning the novel class weights linear relating to features will actually drop the performance. This illustrates the importance of adapting features of novel classes. By AFD, we get consistent and essential performance improvement: 1.73% and 0.51% for 1-shot and 5-shot with mini-ImageNet trained feature extractor, 1.08% and 0.27% for 1-shot and 5-shot with tiered-ImageNet trained feature extractor. This showcases the powerful effects of AFD under cross-domain cases, compared with the simplicity lies in AFD.

6 Conclusion

We propose an finetuning on adaptive feature distribution to improve the novel class generalization for few-shot learning. And the performance improvement on both in-domain and cross-domain evaluation showcases the superior generalization brought by this simple yet effective method. With the proposed AFD method, we also address the importance of further understanding and analyzing the feature distribution of novel classes.

References

Luca Bertinetto, Joao F Henriques, Philip HS Torr, and Andrea Vedaldi. Meta-learning with differentiable closed-form solvers. *arXiv preprint arXiv:1805.08136*, 2018.

Wei-Yu Chen, Yen-Cheng Liu, Zsolt Kira, Yu-Chiang Frank Wang, and Jia-Bin Huang. A closer look at few-shot classification, 2019.

Yinbo Chen, Xiaolong Wang, Zhuang Liu, Huijuan Xu, and Trevor Darrell. A new meta-baseline for few-shot learning. *arXiv preprint arXiv:2003.04390*, 2020.

David L Davies and Donald W Bouldin. A cluster separation measure. *IEEE transactions on pattern analysis and machine intelligence*, (2):224–227, 1979.

Nikita Dvornik, Cordelia Schmid, and Julien Mairal. Diversity with cooperation: Ensemble methods for few-shot classification. *arXiv preprint arXiv:1903.11341*, 2019.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 1126–1135. JMLR. org, 2017.

Victor Garcia and Joan Bruna. Few-shot learning with graph neural networks. *arXiv preprint arXiv:1711.04043*, 2017.

Spyros Gidaris and Nikos Komodakis. Dynamic few-shot visual learning without forgetting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4367–4375, 2018.

Spyros Gidaris and Nikos Komodakis. Generating classification weights with gnn denoising autoencoders for few-shot learning. *arXiv preprint arXiv:1905.01102*, 2019.

Spyros Gidaris, Andrei Bursuc, Nikos Komodakis, Patrick Pérez, and Matthieu Cord. Boosting few-shot visual learning with self-supervision. *CoRR*, abs/1906.05186, 2019. URL `http://arxiv.org/abs/1906.05186`.

Yiluan Guo and Ngai-Man Cheung. Attentive weights generation for few shot learning via information maximization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13499–13508, 2020.

Bharath Hariharan and Ross Girshick. Low-shot visual recognition by shrinking and hallucinating features. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 3018–3027, 2017.

Nathan Hilliard, Lawrence Phillips, Scott Howland, Artëm Yankov, Courtney D Corley, and Nathan O Hodas. Few-shot learning with metric-agnostic conditional embeddings. *arXiv preprint arXiv:1802.04376*, 2018.

Ruibing Hou, Hong Chang, MA Bingpeng, Shiguang Shan, and Xilin Chen. Cross attention network for few-shot classification. In *Advances in Neural Information Processing Systems*, pp. 4005–4016, 2019.

Jongmin Kim, Taesup Kim, Sungwoong Kim, and Chang D Yoo. Edge-labeling graph neural network for few-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 11–20, 2019.

Kwonjoon Lee, Subhransu Maji, Avinash Ravichandran, and Stefano Soatto. Meta-learning with differentiable convex optimization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 10657–10665, 2019.

Aoxue Li, Weiran Huang, Xu Lan, Jiashi Feng, Zhenguo Li, and Liwei Wang. Boosting few-shot learning with adaptive margin loss. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12576–12584, 2020.

Hongyang Li, David Eigen, Samuel Dodge, Matthew Zeiler, and Xiaogang Wang. Finding task-relevant features for few-shot learning by category traversal. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–10, 2019a.

Xinzhe Li, Qianru Sun, Yaoyao Liu, Qin Zhou, Shibao Zheng, Tat-Seng Chua, and Bernt Schiele. Learning to self-train for semi-supervised few-shot classification. In *Advances in Neural Information Processing Systems*, pp. 10276–10286, 2019b.

Boris Oreshkin, Pau Rodríguez López, and Alexandre Lacoste. Tadam: Task dependent adaptive metric for improved few-shot learning. In *Advances in Neural Information Processing Systems*, pp. 721–731, 2018.

Hang Qi, Matthew Brown, and David G Lowe. Low-shot learning with imprinted weights. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5822–5830, 2018.

Rajeev Ranjan, Carlos D Castillo, and Rama Chellappa. L2-constrained softmax loss for discriminative face verification. *arXiv preprint arXiv:1703.09507*, 2017.

Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. 2016.

Avinash Ravichandran, Rahul Bhotika, and Stefano Soatto. Few-shot learning with embedded class models and shot-free meta training. *arXiv preprint arXiv:1905.04398*, 2019.

Mengye Ren, Eleni Triantafillou, Sachin Ravi, Jake Snell, Kevin Swersky, Joshua B Tenenbaum, Hugo Larochelle, and Richard S Zemel. Meta-learning for semi-supervised few-shot classification. *arXiv preprint arXiv:1803.00676*, 2018.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. doi: 10.1007/s11263-015-0816-y.

Andrei A Rusu, Dushyant Rao, Jakub Sygnowski, Oriol Vinyals, Razvan Pascanu, Simon Osindero, and Raia Hadsell. Meta-learning with latent embedding optimization. *arXiv preprint arXiv:1807.05960*, 2018.

Christian Simon, Piotr Koniusz, and Mehrtash Harandi. Projective subspace networks for few-shot learning. 2018.

Christian Simon, Piotr Koniusz, Richard Nock, and Mehrtash Harandi. Adaptive subspaces for few-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4136–4145, 2020.

Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems*, pp. 4077–4087, 2017.

Qianru Sun, Yaoyao Liu, Tat-Seng Chua, and Bernt Schiele. Meta-transfer learning for few-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 403–412, 2019.

Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. Learning to compare: Relation network for few-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1199–1208, 2018.

Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. In *Advances in neural information processing systems*, pp. 3630–3638, 2016.

C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The Caltech-UCSD Birds-200-2011 Dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011.

Feng Wang, Xiang Xiang, Jian Cheng, and Alan Loddon Yuille. Normface: l 2 hypersphere embedding for face verification. In *Proceedings of the 25th ACM international conference on Multimedia*, pp. 1041–1049. ACM, 2017.

Yu-Xiong Wang, Ross Girshick, Martial Hebert, and Bharath Hariharan. Low-shot learning from imaginary data. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7278–7286, 2018.